CLAIMS

Please amend the claims as follows:

1. (canceled)

2. (canceled)

3. (currently amended) A processor, comprising:

a plurality of registers;

instruction processing circuitry that fetches an instruction sequence for execution, said instruction sequence including a load instruction and a preceding instruction that precedes said load instruction in program order, wherein said instruction processing circuitry ~~and~~, after fetching said instruction sequence for execution and prior to dispatching said load instruction for execution and responsive to detecting said load instruction within said fetched instruction sequence, translates said load instruction into separately executable prefetch and register operations; and

execution circuitry that performs at least said prefetch operation out-of-order with respect to said preceding instruction to prefetch data and subsequently separately executes said register operation to place said data into a register among said plurality of registers specified by said load instruction.

4. (original) The processor of Claim 3, wherein said execution circuitry executes said register operation in-order with respect to said preceding instruction.

5. (original) The processor of Claim 3, wherein said execution circuitry executes said register operation out-of-order with respect to said preceding instruction.

6. (original) The processor of Claim 3, wherein said prefetch operation and said register operation have a same operation code.

A2

7. (currently amended) The processor of Claim 6, wherein said prefetch operation and said register operation specify a same target register for said data and differ only in a value of a register operation field.

8. (original) The processor of Claim 3, wherein said execution circuitry stores said data prefetched in response to said prefetch operation in a temporary register.

9. (original) The processor of Claim 3, and further comprising a data hazard detector that, in response to detection of a hazard for said data, signals said processor to discard said data and said register operation.

10. (currently amended) A method of performing a load operation in a processor having a plurality of registers, said method comprising:

fetching an instruction sequence for execution, said instruction sequence including a load instruction and a preceding instruction that precedes said load instruction in program order;

in response to fetching said instruction sequence for execution and prior to execution of said load instruction, instruction processing circuitry detecting said load instruction within said fetched instruction sequence and translating said load instruction into separately executable prefetch and register operations;

performing at least said prefetch operation out-of-order with respect to said preceding instruction to prefetch data; and

thereafter, separately executing said register operation to place said data into a register among said plurality of registers specified by said load instruction.

11. (original) The method of Claim 10, and further comprising executing said register operation in-order with respect to said preceding instruction.

12. (original) The method of Claim 10, and further comprising executing said register operation out-of-order with respect to said preceding instruction.

A2

13. (currently amended) The method of Claim 10, wherein translating said load instruction comprises translating said load instruction ~~operation~~ into prefetch and register operations having ~~have~~ a same operation code.

14. (currently amended) The method of Claim 13, wherein said prefetch operation and said register operation specify a same target register for said data and differ only in a value of a register operation field.

15. (original) The method of Claim 10, wherein performing said prefetch operation comprises storing said data in a temporary register.

16. (original) The method of Claim 10, and further comprising:

detecting a data hazard for said data; and

in response to detection of said hazard for said data, discarding said data and said register operation.

17. (new) The processor of Claim 3, wherein said execution circuitry performs said prefetch operation by calculating a speculative target memory address utilizing contents of at least one register without regard for whether said contents will be modified between calculation of said speculative target memory address and performing said register operation and by thereafter initiating a fetch of said data from a memory location associated within said speculative target memory address.

18. (new) The method of Claim 10, wherein performing said prefetch operation comprises:

calculating a speculative target memory address utilizing contents of at least one register without regard for whether said contents will be modified between calculation of said speculative target memory address and performing said register operation; and

thereafter initiating a fetch of said data from a memory location associated within said speculative target memory address.